The MusIC method: a fast and quasi-optimal solution to the muscle forces estimation problem

Antoine Muller, Charles Pontonnier, Georges Dumont

To appear in *Computer Methods in Biomechanics and Biomedical Engineering*
Vol. 00, No. 00, Month 20XX, 1–16

The MusIC method: a fast and quasi-optimal solution to the muscle forces estimation problem

A. Muller[a,*], C. Pontonnier[a,b], G. Dumont[a]

[a] *Univ Rennes, CNRS, Inria, IRISA - UMR 6074, F-35000 Rennes, France*;
[b] *Écoles de Saint-Cyr Coëtquidan, 56380 Guer, France*

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

The present paper aims at presenting a fast and quasi-optimal method of muscle forces estimation: the MusIC method. It consists in interpolating a first estimation in a database generated offline thanks to a classical optimization problem, and then correcting it to respect the motion dynamics. Three different cost functions – two polynomial criteria and a min/max criterion – were tested on a planar musculoskeletal model. The MusIC method provides a computation frequency approximately ten times higher compared to a classical optimization problem with a relative mean error of 4% on cost function evaluation.

Keywords: Musculoskeletal simulation; Computation time; Optimality; Interpolation; Correction

1. Introduction

Musculoskeletal simulation is evolving quickly and numerous software propose musculoskeletal simulations based on inverse dynamics based method (Damsgaard et al. (2006); Delp et al. (2007)) or on EMG-driven models (Buchanan et al. (2004, 2005); Sartori et al. (2012)), even in real time (Manal et al. (2002); Murai et al. (2010); van den Bogert et al. (2013)). In such software using inverse dynamics based methods, the force sharing problem is solved thanks to optimization methods. In An et al. (1984) and Herzog (1987), the force sharing problem is assumed to be an optimization problem, consisting in minimizing a criterion representing a central nervous system (CNS) strategy. The criterion represents a cost, e.g. metabolic energy, muscle fatigue or joint reaction force (Crowninshield and Brand (1981); Challis (1997); Rasmussen et al. (2001); Dumas et al. (2014)).

However, optimization remains costly in terms of computation time, despite of several implementations and improvements in the last years. Mostly, the use of Sequential Quadratic Programming Methods (SQP) have deeply improved the computation times since the muscle forces estimation problem is well shaped for such an algorithm.

However, in real-time simulations including muscle forces estimation, the result remains sub-optimal. In the method proposed by Murai et al. (2010), muscles were gathered by functional groups to reduce the problem complexity, that led to strong bias in the estimated forces. In the work of van den Bogert et al. (2013), the use of a neural network dedicated to quadratic optimization led to a real-time but sub-optimal result, since computation time was limited to ensure real-time computation. Moreover, force-length and force-velocity relationships were not taken into account in the muscle models in this paper.

Interpolation has been proposed as a solution to reduce the computation time (Pontonnier and Dumont (2009, 2010)). Although barycentric interpolation of muscle forces based on a database

*Corresponding author. Email: antoine.muller@irisa.fr

of optimized forces led to a faster solution than classical optimization, the method did not ensure the dynamical equilibrium of the interpolated forces. Moreover, the database remained difficult to extend to multibody models and multi-articular muscles, mostly due to the high degree of coupling existing between joints.

The current study aims at presenting a fast and quasi-optimal method of muscle forces estimation, the MusIC method (Muscle forces Interpolation and Correction) and its performance in terms of mean computation frequency and accuracy. The next section presents the MusIC method and the application developed to compare this method with a classical SQP method. Results are presented and discussed in terms of optimality with regard to the chosen cost function and external forces applied to the system, similarity between the solutions found by both methods and computation time.

2. Methods

In the following section, we consider a generic musculoskeletal model composed of n_b bodies, articulated by n_j joints and actuated by n_m muscles.

From the musculoskeletal model geometry, we construct the $(n_j \times n_m)$ matrix $\boldsymbol{M}$ which contains the analytical expressions of the moment arms of each muscle for each joint with respect to joint coordinates. For a given frame, this matrix is numerically evaluated from joint coordinates values. $\boldsymbol{F}$ refers to the $(n_m \times 1)$ muscle forces vector at a given frame. $\boldsymbol{F_{max}}$ is the $(n_m \times 1)$ maximal muscle forces vector containing the maximal physiological force that can be provided by each muscle. The input of the MusIC method are the joint coordinates and the joint torques, issued from motion. Thus, the $(n_j \times 1)$ vector $\boldsymbol{q}$ contains the joint coordinates and the $(n_j \times 1)$ vector $\boldsymbol{\Gamma}$ contains the joint torques.

2.1 *MusIC method overview*

The MusIC method is based on two main hypotheses:

- the muscle forces problem can be first solved joint per joint and the inter-joint muscular coupling (multi-articular muscles) can be taken into account a posteriori;
- the muscle forces can be corrected to respect the dynamic equilibrium.

Therefore, the MusIC method is separated in two stages. The first one consists in computing a database describing the muscle forces sharing solution joint per joint. The second one consists in interpolating forces thanks to this database, mixing them to take into account muscular coupling and finally correcting them to respect the dynamics of the motion.

The database contains the classical muscle force sharing solutions – solutions of the static optimization under constraints of a cost function representing the CNS behavior –, stored in an $(n_m \times 1)$ activation ratio vector $\boldsymbol{\alpha}$, solution of the force sharing problem for the considered joint. It corresponds to the activity of each muscle normalized by the total activity of the muscles involved in this joint's motion (1). By considering only one joint, it allows muscle forces to be computed independently from the torque applied to generate the database. Indeed, the values of $\boldsymbol{\alpha}$ are not dependent of this applied torque due to the normalization by the sum of activations of all muscles.

$$\alpha_m = \frac{F_m/F_{max,m}}{\sum_{i=1}^{n_m} F_i/F_{max,i}} \tag{1}$$

Where m is the m^{th} muscle. The next section details the computation and the storage of the vector $\boldsymbol{\alpha}$ in a generic case.

2.2 *Database computation*

An activation ratio vector database consists in n_j independent sub-databases (left part Figure 1) containing activation ratios for each joint of the model.

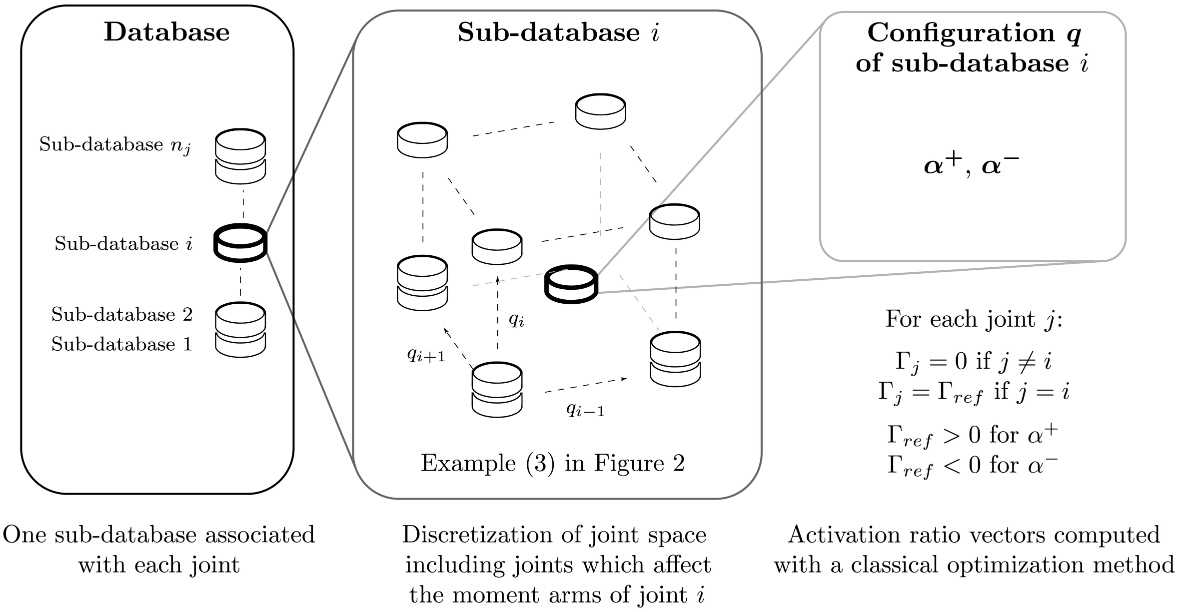

Figure 1. Activation ratio vector databases structure, composed of n_j independent sub-databases (left part). Each sub-database gathers $\boldsymbol{\alpha}$ for different joint configurations (central part). For each configuration, $\boldsymbol{\alpha}$ are computed and stored (right part).

Each sub-database is composed of activation ratio vectors $\boldsymbol{\alpha}$. As they depend on moment arms, a sub-database stores $\boldsymbol{\alpha}$ for different joint configurations (central part Figure 1). For the ith sub-database, only joints affecting moment arms of joint i are considered. To handle this, a symmetric $(n_j \times n_j)$ muscular coupling matrix $\boldsymbol{C}$ is associated to the musculoskeletal model. C_{ij} is 1 if there is at least one muscle coupling joints i and j, zero otherwise. Figure 2 shows three examples of muscular coupling: mono-articular, bi-articular and tri-articular muscle and the corresponding coupling terms. The dimension of sub-database i is the number of joints affecting the moment arms of joint i. Joint space of database i is discretized to generate joint configurations. For each of these sub-database configurations, the other joints coordinates are set to zero. The central part in Figure 1 represents the 3-dimension sub-database i considered in the example (3) in Figure 2.

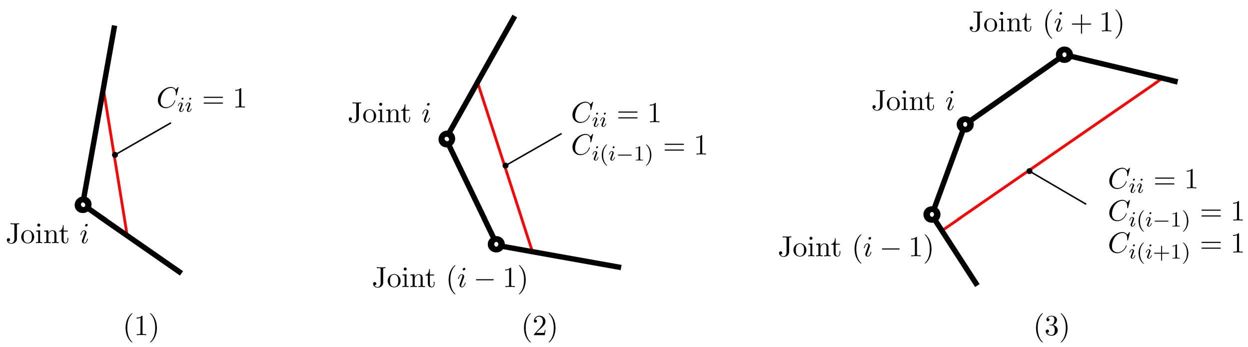

Figure 2. Example of different muscular coupling. (1): mono-articular muscle actuating the joint i. (2): bi-articular muscle actuating joints $(i-1)$ and i. (3): tri-articular muscles actuating joints $(i-1)$, i and $(i+1)$. Black lines represent segments and black circles represent revolute joints. Red lines represent muscles attached to the segments.

For each configuration, activation ratio vectors are computed thanks to an optimization problem (2) and then stored in the database (right part Figure 1). It consists in minimizing a cost function while respecting the dynamic equilibrium and the positivity of forces exerted by muscles due to the fact that muscles can only pull. The maximal muscle forces are not taken into account in this optimization problem since only ratios between forces are stored in the database. A reference torque Γ_{ref} is assigned to joint i and the other joints torques are set to zero value. The muscle force sharing solution depends on the sign of the joint torque, therefore the algorithm computes two separate groups of $\boldsymbol{\alpha}$: $\boldsymbol{\alpha}^+$ for a positive reference torque ($\Gamma_{ref} > 0$) and $\boldsymbol{\alpha}^-$ for a negative reference torque ($\Gamma_{ref} < 0$).

$$\begin{aligned} \underset{\widetilde{\boldsymbol{F}}}{\text{minimize}} \quad & f(\widetilde{\boldsymbol{F}}) \\ \text{subject to} \quad & \widetilde{\boldsymbol{M}}\widetilde{\boldsymbol{F}} = \widetilde{\boldsymbol{\Gamma}}_{\boldsymbol{ref}}, \\ & \widetilde{\boldsymbol{F}} \geq \boldsymbol{0}. \end{aligned} \tag{2}$$

Where $\sim$ refers to the joints affecting the moment arms of the joint associated to this sub-database – obtained from the coupling matrix $\boldsymbol{C}$ – and muscles actuating this joint. Thus, $\widetilde{\boldsymbol{M}}$ is the sub-matrix of $\boldsymbol{M}$ containing the rows associated to the referred joints and the columns associated to the referred muscles. $\widetilde{\boldsymbol{F}}$ is the sub-vector of $\boldsymbol{F}$ containing the forces associated to the referred muscles. $\widetilde{\boldsymbol{\Gamma}}_{\boldsymbol{ref}}$ contains the torques applied in referred joints – Γ_{ref} for the joint associated to the sub-database and zero value for the others. $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ are then computed thanks to the equation (1) by using $\widetilde{\boldsymbol{F}}$ and by assigning a zero force for the joints that are not considered in $\widetilde{\boldsymbol{F}}$.

2.3 *Muscle forces estimation*

From an experimental or a simulated motion, the MusIC method aims at estimating the muscle forces. Thanks to joint coordinates $\boldsymbol{q}$ and joint torques $\boldsymbol{\Gamma}$, the method uses the previously generated database to compute a muscle forces estimation frame per frame. It consists in two global steps: an interpolation step and a correction step (Figure 3). From the current joint coordinates and joint torques, a first estimation of the muscle forces for each joint ($\boldsymbol{F}^{(1)}$, $\boldsymbol{F}^{(2)}$, ..., $\boldsymbol{F}^{(n_j)}$) – set of $(n_m \times 1)$ vectors – is interpolated from the activation ratio vector database. Then, the correction step find a solution $\boldsymbol{F}$ close to the interpolation result that respects the dynamic equilibrium and the physiological properties.

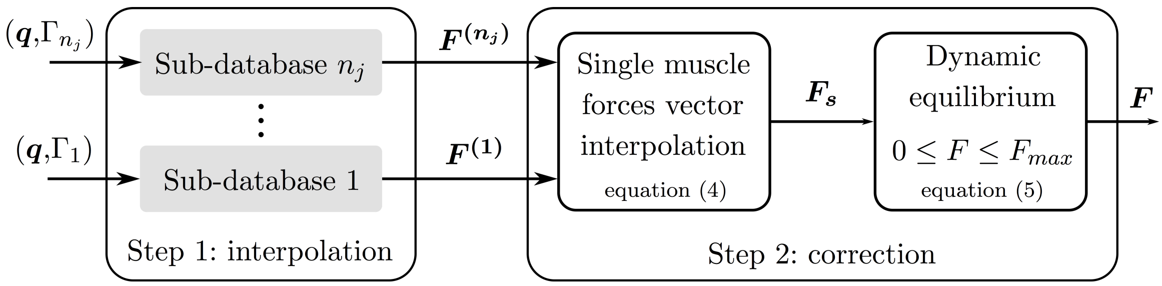

Figure 3. MusIC method pipeline. Muscle forces are deducted from joint coordinates $\boldsymbol{q}$ and joint torques $\boldsymbol{\Gamma}$. A first estimation is interpolated from the database for each joint. The correction step finds a solution $\boldsymbol{F}$ close to the first estimation that respects the dynamic equilibrium and the physiological properties.

Step 1 computes muscle forces that respect the joint torques and the activation ratio vector for each joint configuration (3). For joint i, the interpolated activation ratio vector $\boldsymbol{\alpha}^{(i)}_{\boldsymbol{interp}}$ is extracted from the sub-database i thanks to a linear interpolation between the stored values. The sign of Γ_i decides to use $\boldsymbol{\alpha}^+$ or $\boldsymbol{\alpha}^-$.

$$\boldsymbol{F}^{(i)} = \left(\frac{\Gamma_i}{\boldsymbol{M_{i*}} \cdot (\boldsymbol{\alpha}^{(i)}_{\boldsymbol{interp}} \circ \boldsymbol{F_{max}})} \right) \left(\boldsymbol{\alpha}^{(i)}_{\boldsymbol{interp}} \circ \boldsymbol{F_{max}} \right) \tag{3}$$

Where $\boldsymbol{M_{i*}}$ corresponds to the ith row of the moment arms matrix $\boldsymbol{M}$ and $\circ$ is the entrywise product.

Step 2 gathers and corrects forces obtained at step 1. Step 1 gives several forces values for a single muscle (one per joint on which the muscle acts). Thus, step 2 first computes a single muscle force for each muscle, gathered in a $(n_m \times 1)$ vector $\boldsymbol{F_s}$. For each muscle, the force value is computed with a barycentric interpolation of all the forces associated to the muscle (4). The torques associated with each joint are used as weights in the interpolation.

$$F_{s,m} = \frac{\sum_{i \in J_m} F_m^{(i)} \; \Gamma_i}{\sum_{i \in J_m} \Gamma_i} \tag{4}$$

where J_m is the list of joints actuated by the muscle m. Once $\boldsymbol{F_s}$ is obtained, we solve the optimization problem (5). It consists in finding the closest solution to $\boldsymbol{F_s}$ subject to the dynamic equilibrium and the physiologicial properties – muscles can only pull with maximal forces. An active set method using Karush-Kuhn-Tucker (KKT) conditions and directly implemented from Dumont (1995) is used to solve this problem. The gradients are analytically computed since the cost function is quadratic and constraint equations are linear.

$$\begin{aligned} \underset{\boldsymbol{F}}{\text{minimize}} \quad & \sum_{i=1}^{n_m} \left(\frac{F_i - F_{s,i}}{F_{max,i}} \right)^2 \\ \text{subject to} \quad & \boldsymbol{MF} = \boldsymbol{\Gamma}, \\ & \boldsymbol{0} \leq \boldsymbol{F} \leq \boldsymbol{F_{max}}. \end{aligned} \tag{5}$$

Finally, the vector $\boldsymbol{F}$ is the solution of the muscle force estimation problem for this frame.

3. Application

The method was applicated and compared to a classical optimization method for a large set of motions applied to a simple but representative musculoskeletal model.

3.1 *Model*

A transverse planar shoulder and elbow model was used. The skeletal system consisted of the thorax – considered fixed –, the arm and the forearm (Figure 4). Geometrical and inertial parameters were obtained respectively from Dempster (1955) and Dumas et al. (2007) and uniformly scaled to the subject's size and mass. One synthetic morphology was used to test the method: 180 cm/80 kg.

The muscular geometry was based on the virtual arm model developed by Song et al. (2008) and consisted in 12 muscles as detailed in Table 1. The maximum force produced by a muscle was defined from the maximum isometric force f_0 and the muscle length – depending on the joint coordinates $\boldsymbol{q}$ – by using the length-force relation proposed by Rengifo et al. (2010) (Figure 5).

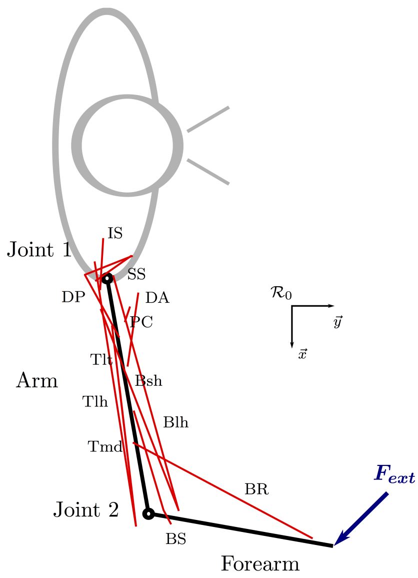

Figure 4. Schematic representation of the transverse planar arm model used in this study. The same representation as in Figure 2 is used. Broken lines illustrate muscles with via points associated to the segments.

Muscles		Joint actuated by this muscle		f_0 (N)
		Shoulder	Elbow	
DA	Deltoid anterior	x		1,148
DP	Deltoid posterior	x		266
PC	Clavicular portion of pectoralis major	x		490
SS	Supraspinatus	x		1,203
IS	Infraspinatus	x		364
Blh	Biceps long head	x	x	630
Bsh	Biceps short head	x	x	434
Tlh	Triceps long head	x	x	611
Tlt	Triceps lateral head		x	763
Tmd	Triceps medial head		x	630
BS	Brachialis		x	994
BR	Brachioradialis		x	266

Table 1. Muscles in the model. f_0 is the maximum isometric force.

An external constant force $\boldsymbol{F_{ext}}$ was applied to the distal point of the forearm, with four different levels of intensity [0 0], [0 50], [50 0] and [50 50] – expressed in the global frame $\mathcal{R}_0$ –.

3.2 *Databases generation*

In this application, to test the robustness of the method, three different cost functions were tested: two polynomial criteria (6)(7) (Pedotti et al. (1978); Herzog (1987); Happee (1994)) and one min/max criterion (8) (Rasmussen et al. (2001)). The behaviors induced by these usual cost functions are different. Indeed, the two polynimal criteria used tend to recruit in priority strongest muscles whereas the min/max criterion tends to recruit in similar proportion all the synergistic muscles. Thus, three databases associated to the three cost functions were generated as described in Section 2.2.

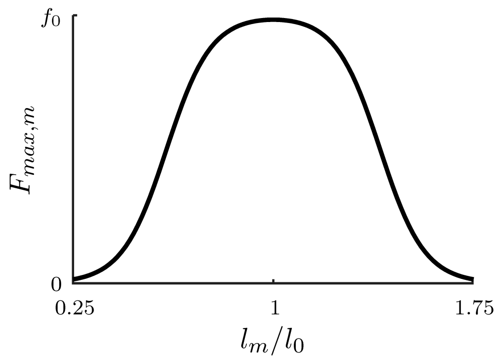

Figure 5. Force-length relationship (Rengifo et al. (2010)). $F_{max,m}$ is the maximum force produced by a muscle, f_0 the maximum isometric force, l_m the current length – depending on the joint coordinates $\boldsymbol{q}$ – and l_0 the optimal fiber length.

$$f(\boldsymbol{F}) = f_1(\boldsymbol{F}) = \sum_{m=1}^{n_m} \left(\frac{F_m}{F_{max,m}} \right)^2 \tag{6}$$

$$f(\boldsymbol{F}) = f_2(\boldsymbol{F}) = \sum_{m=1}^{n_m} \left(\frac{F_m}{F_{max,m}} \right)^3 \tag{7}$$

$$f(\boldsymbol{F}) = f_3(\boldsymbol{F}) = \underset{m \in [\![1:n_m]\!]}{max} \left(\frac{F_m}{F_{max,m}} \right) \tag{8}$$

Cost functions f_1 and f_2 were directly used in optimization problem (2). As proposed by Rasmussen et al. (2001), the optimization problem with f_3 was reformulated as a bound formulation (9) to improve the numerical resolution. Optimization problems were solved with a SQP method (*fmincon* function of Matlab®).

$$\begin{aligned} \underset{\beta,\boldsymbol{F}}{\text{minimize}} \quad & \beta \\ \text{subject to} \quad & \boldsymbol{F} \leq \beta, \\ & \boldsymbol{MF} = \boldsymbol{\Gamma}, \\ & \boldsymbol{F} \geq \boldsymbol{0}. \end{aligned} \tag{9}$$

After a preliminary study, we chose to discretize the accessible joint space of each joint in 60 parts between $-180°$ and $180°$. This value ensured results convergence with an acceptable computation time – about one hour – for the database generation. In this example, with bi-articular muscles, the muscular coupling matrix $\boldsymbol{C}$ is a (2×2) matrix and each sub-database contained 3,600 (60×60) $\boldsymbol{\alpha}$ entries.

3.3 *Motions generation*

To compare the MusIC method to a classical method, a set of synthetic unconstrained point-to-point motions was chosen. To generate a set of 36 start-end points (forearm distal point), we chose

6 joint coordinates equally distributed between 5° and 90° for the shoulder and 6 joint coordinates equally distributed between 5° and 90° for the elbow. All point-to-point combinations between the 36 points were generated. All the motions were completed successively in 0.5 second and in 1 second, sampled at 100 Hz, representing finally 2,520 motions.

As mentioned before, the input of the MusIC method are the joint coordinates and the joint torques. Thus, joint trajectories were obtained thanks to the work of Flash and Hogan (1985), minimizing the mean-square jerk. At each frame, the joint coordinates vector $\boldsymbol{q}$ is thus extracted. Joint torques $\boldsymbol{\Gamma}$ were computed thanks to a recursive Newton-Euler algorithm – based on spatial algebra – (Featherstone (2008)) taking into account the external force $\boldsymbol{F_{ext}}$. For this end, for each motion, trajectories $\boldsymbol{q}$ were numerically differentiated to obtain joint velocities and accelerations. The gravity was assumed to be zero, since the arm worked in the transverse plane as in the Flash and Hogan (1985) experimentation.

3.4 *Muscle forces estimation*

Muscles forces $\boldsymbol{F}$, computed thanks to the MusIC method, and reference muscle forces $\boldsymbol{F_{ref}}$, computed thanks to a classical optimization method, were compared.

3.4.1 MusIC method

According to the cost function, muscle forces $\boldsymbol{F}$ were computed in accordance with the method described in section 2.

3.4.2 Classical optimization method

Reference muscle forces $\boldsymbol{F_{ref}}$ were computed by solving an optimization problem (10) with each cost function $f(\boldsymbol{F_{ref}})$ and each motion. Constraint functions correspond to the dynamic equilibrium and the physiological properties. For the min/max optimization problem, a bound formulation was used (Rasmussen et al. (2001)). Optimization problems were classically solved by applying a SQP method.

$$\begin{aligned} \underset{\boldsymbol{F_{ref}}}{\text{minimize}} \quad & f(\boldsymbol{F_{ref}}) \\ \text{subject to} \quad & \boldsymbol{M}\boldsymbol{F_{ref}} = \boldsymbol{\Gamma}, \\ & \boldsymbol{0} \leq \boldsymbol{F_{ref}} \leq \boldsymbol{F_{max}}. \end{aligned} \tag{10}$$

3.4.3 Comparison

Both methods were compared in terms of accuracy and computation time. During the resolution of the optimization problem 5, the algorithm did not converge for some configurations whereas all agonist or antagonist muscles provide their maximal force. Thus, these configurations were not physiologically plausible due to the fact that the motions were synthetically generated. Motions containing one of these configurations were detected and not taken into account in the statistics. For each motion, a cost function matching indicator ϵ_f (11) averaged over the motion (over-lined variables) was evaluated to assess the match between the cost function values computed by using $\boldsymbol{F_{ref}}$ and computed by using $\boldsymbol{F}$.

$$\epsilon_f = \overline{\frac{f(\boldsymbol{F}) - f(\boldsymbol{F_{ref}})}{\max(f(\boldsymbol{F_{ref}}))}} \tag{11}$$

Moreover, cross-correlation was used to compare the muscle forces obtained by both methods in terms of shape. Last, the mean computation frequency of both methods were collected by dividing the whole computation time by the number of frames. Only the computation time of muscle forces estimation parts were considered. We chose to use the mean computation frequency to allow motions with different duration to be compare.

4. Results and discussion

The first step consisted in detecting and removing unfeasible motions. Thus, considering all combinations of motions, external forces and cost functions, we kept 18,207 motions on the 25,920 generated ones.

Globally, the mean cost function matching indicator was $4 \pm 3.7\%$ for all trials. Table 2 shows the mean cross-correlation results for each muscle force. The mean cross-correlation coefficients ranged between 0.89 and 0.99. The maximal mean time delay was 0.16 seconds. For the classical optimization method and for the MusIC method, the mean computation frequencies were respectively 108 ± 33 Hz and 1000 ± 180 Hz.

	Muscles	r	τ(s)
DA	Deltoid anterior	0.99	-0.14
DP	Deltoid posterior	0.93	-0.05
PC	Clavicular portion of pectoralis major	0.98	-0.11
SS	Supraspinatus	0.94	-0.02
IS	Infraspinatus	0.95	-0.05
Blh*	Biceps long head*	0.96	-0.07
Bsh*	Biceps short head*	0.92	-0.03
Tlh*	Triceps long head*	0.91	-0.03
Tlt	Triceps lateral head	0.90	-0.16
Tmd	Triceps medial head	0.90	-0.16
BS	Brachialis	0.90	-0.02
BR	Brachioradialis	0.89	-0.02

Table 2. Mean cross-correlation coefficient (r) and mean time lag (τ) between muscle forces obtained with both methods. Bi-articular muscles are noted with a *.

The cost function matching indicators distributions differ according to the cost function used (Figure 6). The ϵ_f followed a normal distribution (verified by a quantile-quantile linear correlation), thus a f-test was performed between cost function results. Significant differences were detected between each cost function results ($p < 0.001$). The mean cost function matching indicators were $1.85 \pm 1.70\%$, $2.52 \pm 2.30\%$ and $7.12 \pm 3.65\%$ for f_1, f_2 and f_3 respectively.

Figure 7 compares the three cost function values obtained by both methods with respect to time. Motions 1, 2 and 3 exhibit respectively a low ϵ_f, a median ϵ_f and a high ϵ_f compared to the mean. In every case, cost function f_1 and f_3 led to the lowest and highest ϵ_f values respectively. Moreover, the highest absolute differences between the two methods were found where muscle activations were the most important.

For the motion 2 – ϵ_f comparable to the mean –, we analyze the muscle forces obtained by both methods with the cost function f_1 (Figure 8), the cost function f_2 (Figure 9) and the cost function f_3 (Figure 10). In accordance with the cost function matching indicators results, muscle forces obtained considering the cost function f_3 are further from the classical method than the others. This result is particularly visible on the deltoid posterior (DP), on the triceps long head (Tlh) and on the triceps medial head (Tmd). For the two others cost function, results are very similar. For each muscle, the activation or the non-activation is respected with the MusIC method. The

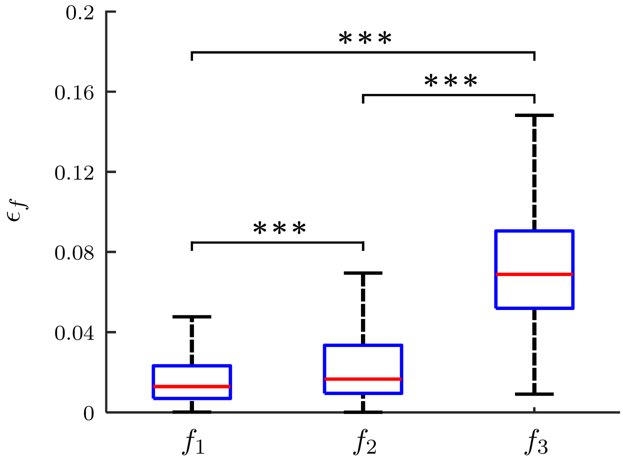

Figure 6. Box plots representing cost function matching indicators distributions for the different cost function used. Significant differences at $p < 0.001$ are denoted ***. Outliers were not displayed.

three muscular chiefs of Triceps exhibit a small difference in amplitude and force distribution. The deltoid posterior muscle (DP) is representative of the recruiting differences between cost functions. This behavior was correctly mimicked by the MusIC method, leading to more muscle synergy with a higher polynomial degree.

Results are encouraging: in any case, the method finds a quasi-optimal solution which respects the dynamic equilibrium and the physiological properties. We consider the solution as quasi-optimal with respect to the classical optimization problem of the considered cost function. Cross-correlations show that each muscle force exhibit a similar shape between methods. The main benefit of this sub-optimal method is the computation frequency which is, in this case, approximately ten times higher. However, the MusIC method requires an additionnal computation time corresponding to the database generation time – about one hour. This time may seem important and widely increase the global computation time. However, several improvements are suggested in the following discussion to decrease the computation time needed to generate the database.

One of principles of this method is to consider separately the muscles contribution for each joint. So, the poly-articular muscles are those which could give the worst results. However, the cross-correlation for these muscle forces seems to be consistent with the others. As the method proposed aims at mimicking a static optimization, it has the same limitations. One of them is the lack of continuity of the solution from one frame to one other. This issue is however pondered by the fact that sampling time is quite small.

Large differences appear according to the cost function used. That means MusIC method does not behave in the same way for each of them. Indeed, errors with cost function f_3 are higher and more variable than with the others where the error on cost function is always less than 7%.

This is also the case in the three motions used as illustrative and representative cases. As mentionned above, cost function f_1 and f_3 led to the lowest and highest ϵ_f values respectively. The illustrative motions also seemed to show that the more the joint motion is important, the more the error is important. Since motion duration is equal in any case, it seems to show that higher accelerations led to less accurate results. However, this is not that sensitive with cost functions f_1 and f_2 as presented on Figure 6.

The results presented above are also confirmed by the muscle forces values. Cost functions f_1 and f_2 led to very similar activations and amplitudes whatever the method used. Only the triceps chiefs revealed a small difference in amplitude. Since these muscles actions were very similar, the MusIC method was not accurate enough to distinguish these contributions properly. Cost function f_3 exhibited similar trends but the difference between methods was higher. Moreover, the activation pattern of muscles was not always respected. Indeed, as expected, min/max criterion

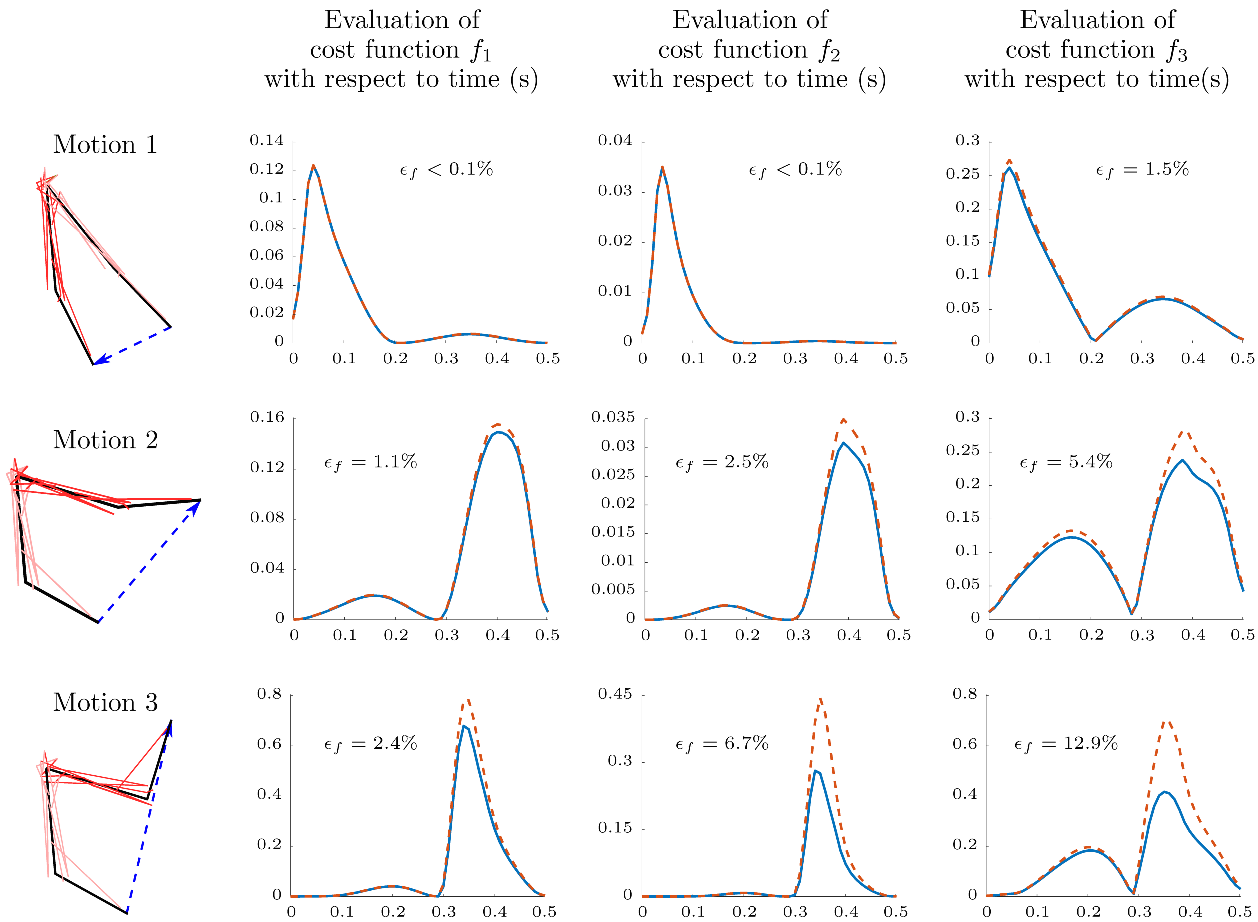

Figure 7. Evolution of the three cost function values with respect to time (in seconds) for three different motions. The continuous blue line represents results obtained with the classical optimization method. The dotted red line represents results obtained with the MusIC method. Each corresponding ϵ_f was noted on graphs. Motions 1,2 and 3 exhibit a low ϵ_f compared to the mean, a median ϵ_f and a high ϵ_f compared to the mean respectively. Each of the three motions were completed in 0.5 second without applied external forces.

could lead to non-smooth muscle activation patterns (An et al. (1984); Crowninshield and Brand (1981); Rasmussen et al. (2001)). The MusIC method has a natural smoothing effect – barycentric interpolation detailed in (4) – that could explain these differences. It is particularly visible on the deltoid anterior (DA), on the deltoid posterior (DP) and on the pectoralis major (PC) where the abrupt changes are not followed by the MusIC method.

Thus, the MusIC method seems to be very beneficial for the polynomial criteria where the observed results are comparable to a classical optimization problem while largely decreasing the computation time. However, errors observed in some conditions with the min/max criterion appear to ask for improvements to use such a method with such a cost function.

Moreover, to stay in accordance with the idea to work with activations, we chose to normalize forces by the maximal muscle forces in the optimization problem (5) where the dynamic equilibrium and the physiological properties are taken into account. This normalization emphasizes muscles with larger maximal forces values.

The method was applicated on a simple but representative musculoskeletal model. However, additional validations have to be achieved with more complex models, real experimental data and numerous subjects. Particularly, we need to validate the method on a three dimensional model.

The use of a more complex musculoskeletal model is mandatory to define more precisely some parameters arbitrarily chosen in this study, particularly for the database generation. Indeed, with increasing degrees of freedom, the databases dimension will increase – for example with a classical

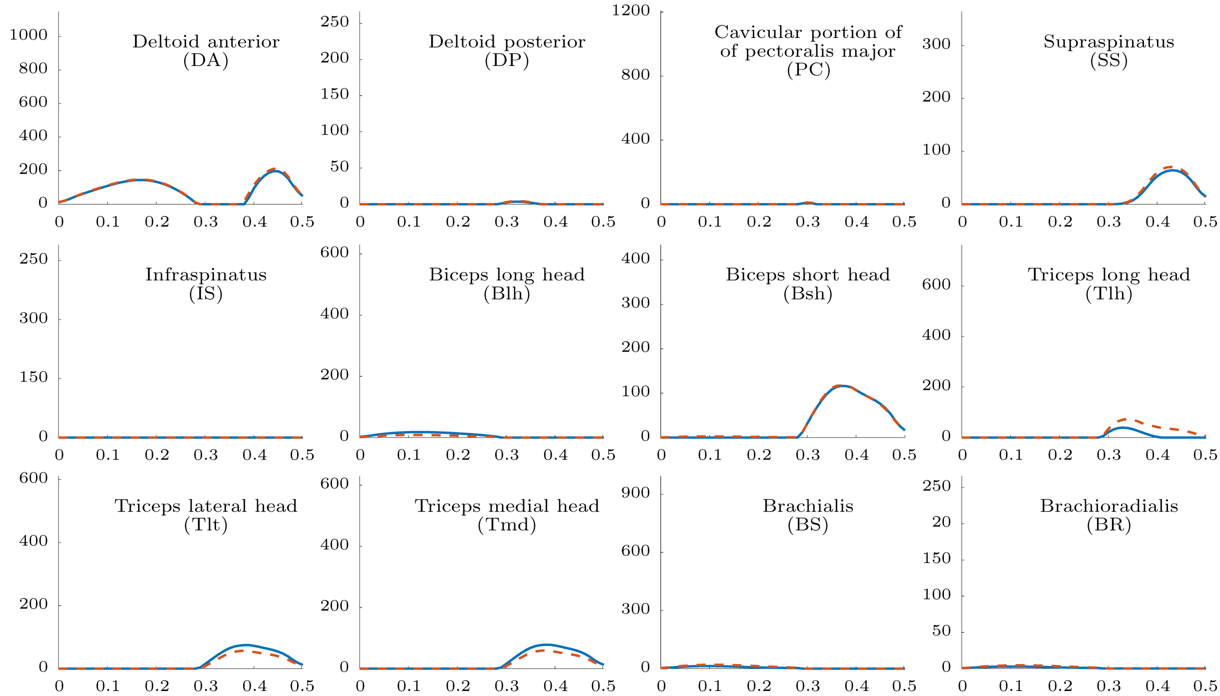

Figure 8. Muscles forces (in Newtons) of the motion 2 (Figure 7) obtained by both methods with the cost function f_1 with respect to time (in seconds). The continuous blue line represents the classical optimization method. The dotted red line represents the MusIC method. Muscle force scale goes from zero to the maximum isometric force (f_0).

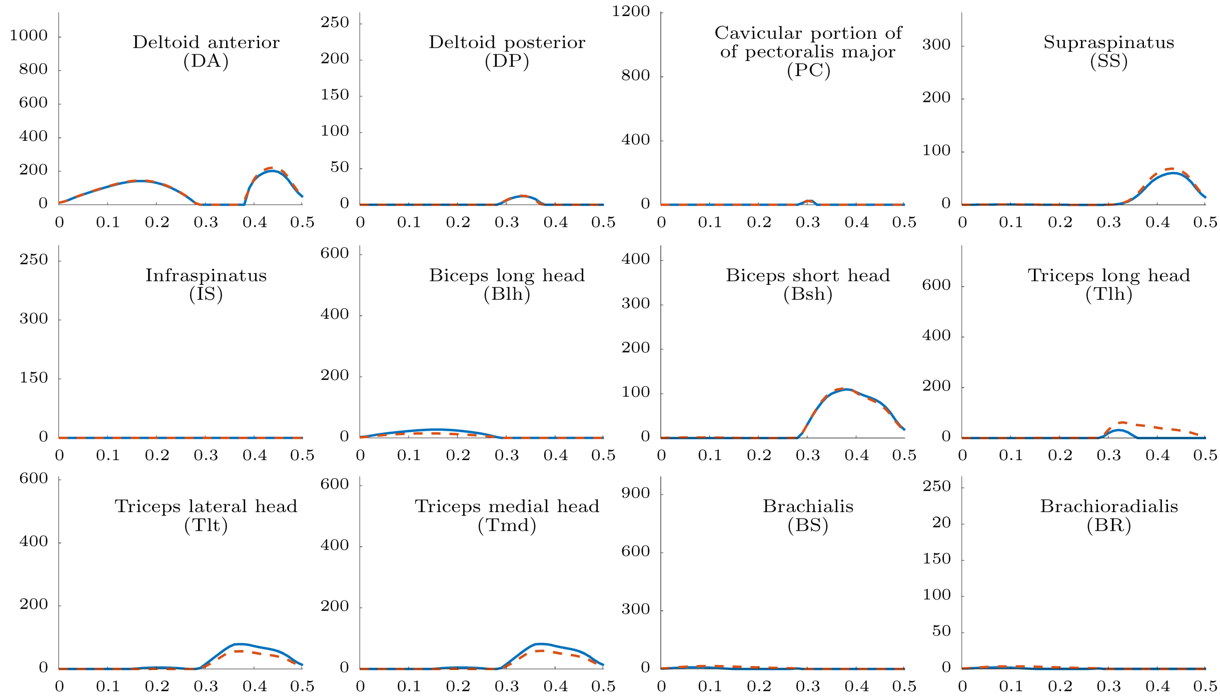

Figure 9. Muscles forces (in Newtons) of the motion 2 (Figure 7) obtained by both methods with the cost function f_2 with respect to time (in seconds). The continuous blue line represents the classical optimization method. The dotted red line represents the MusIC method. Muscle force scale goes from zero to the maximum isometric force (f_0).

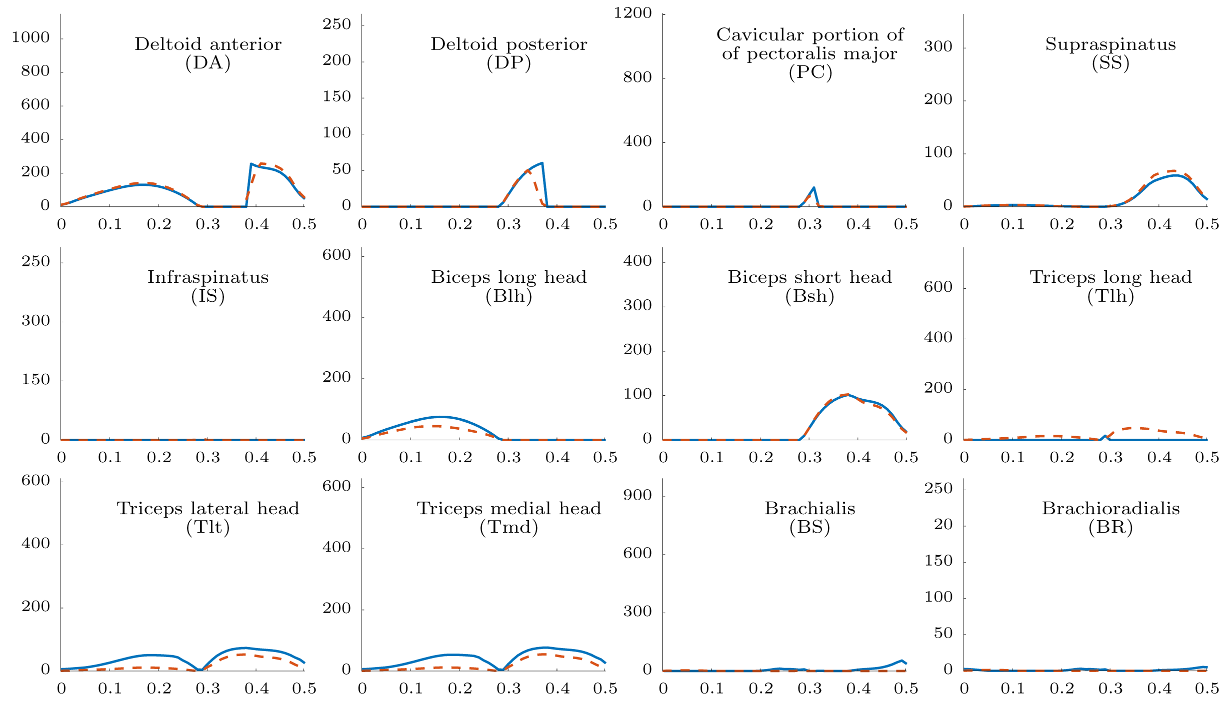

Figure 10. Muscles forces (in Newtons) of the motion 2 (Figure 7) obtained by both methods with the cost function f_3 with respect to time (in seconds). The continuous blue line represents the classical optimization method. The dotted red line represents the MusIC method. Muscle force scale goes from zero to the maximum isometric force (f_0).

musculoskeletal leg model (Klein Horsman et al. (2007)), a sub-database generation could involve six different joints –. This is probably not an issue for the data extraction which is achieved separately for each dimension. However, the time spent on databases generation could become an obstacle.

So it is of importance to find a trade-off between the density of the database and the time spent for its generation. This is a study that will be made in a near future to allow the method to be used with complex three dimensional models with a reasonable database generation time, a study evaluating the accuracy of the method with regard to the database density has to be driven. An other solution could be to define a non-homogeneous database. This solution is able to reduce the density of some databases spaces while refining the results for others. For example, moment arm sign changes are critical and need a high density of the database whereas extreme joint positions seem less sensitive and databases may be less populated at these locations. Such a method can also be used to improve the results consistency, in particular with the min/max criterion.

Currently, one database must be generated for one specific subject with one cost function. Since the time spent for the generation may be significant, a dimensionless database could be generated and used for all subjects with the cost function considered. This dimensionless database could be adapted to the subject's geometrical, inertial and muscular properties thanks to an additional scaling step.

Moreover, to simplify the implementation and to limit the computation time, we chose to use a linear interpolation to extract an activation ratio vector corresponding to the current joint coordinates. It could be interesting to evaluate the influence of the chosen interpolation method on results. Particularly, the most suitable interpolation method can vary according to the cost function used.

The MusIC method was assessed with three different cost functions: two polynomial criteria and one min/max criterion. Other functions usually exploited to evaluate muscle forces could be tested with the MusIC method. It only consists in using a new cost function throughout the database generation. A polynomial criteria with another normalization factor as the physiological cross sectional area (PCSA) (Crowninshield and Brand (1981)) or a soft saturation criteria (Siemienski (1992)) could be tested. These cost function are based on muscle forces or activations. Some authors have proposed to introduce joint reaction forces as a cost (Challis and Kerwin (1993) and Dumas et al. (2014)). Even if the MusIC method does not take into account directly the joint reaction forces, they are associated to the joint torques. We therefore believe that the use of the MusIC method with this kind of cost function could be an interesting future work. We could also generate the database thanks to cost functions generating co-contraction behaviors (Forster et al. (2004); Brookham et al. (2011)). Since the MusIC method aims at mimicking an optimization method, each generated database will lead to a unique solution for a given frame with respect to this cost function. However different databases may be used to generate a variety of solutions, or a unique database containing various solutions.

Moreover, the muscle model used as a application exhibits only a force-length relationship. Another improvement would be to extend the method to muscle models exhibiting force-velocity relationships (Hill (1938); Zajac (1989)). This would lead to change the sub-database constitution by including joint velocities storage.

Finally, the use of joint torques as input data of the method allows only models with open loop chains to be used. With this kind of models, the optimization method used in this paper as the reference is the same as classically used in the litterature (Damsgaard et al. (2006); Delp et al. (2007)). However, some musculoskeletal models which try to improve the anatomy representation by using closed loop chains (Van Der Helm (1994); Maurel et al. (1996); Pennestri et al. (2007)). Thus, to take into account these models, we would have to adapt the MusIC method to not consider joint torques as an input but directly the motion.

5. Conclusion

This paper aimed at presenting a fast and quasi-optimal method of muscle forces estimation: the MusIC method. It consists in interpolating a first estimation in a database generated offline and then correcting it with respect to the motion dynamics. This method was tested on a planar musculoskeletal arm model. Results were compared with a state-of-the-art muscle forces estimation method

For a large set of motions, the method allowed to find a solution which respects the dynamic equilibrium and the physiological properties and corresponded to an average cost function close to the optimal value. The method showed a particularly low computation time compared to the classical SQP method – approximately ten times faster. The method was particularly efficient with polynomial criteria whereas the min/max criterion leaded to less accurate results.

After additional validations, we consider implementing the method in a motion analysis library in the near future. This would lead to a very appealing product, enabling fast and accurate musculoskeletal simulation. Such a feature has a great potential in the development of new usages in rehabilitation, sports or ergonomics. Indeed, the use of this type of method could enable a direct feedback to the user thus improve such setups.

Acknowledgements

This work was partially supported by the ANR project ENTRACTE under Grant ANR 13-CORD-002-01.

References

An K, Kwak B, Chao E, Morrey B. 1984. Determination of muscle and joint forces: a new technique to solve the indeterminate problem. Journal of biomechanical engineering. 106(4):364–367.

Brookham RL, Middlebrook EE, Grewal TJ, Dickerson CR. 2011. The utility of an empirically derived co-activation ratio for muscle force prediction through optimization. Journal of Biomechanics. 44(8):1582–1587.

Buchanan TS, Lloyd DG, Manal K, Besier TF. 2004. Neuromusculoskeletal modeling: Estimation of muscle forces and joint moments and movements from measurements of neural command. Journal of Applied Biomechanics. 20(4):367–395.

Buchanan TS, Lloyd DG, Manal K, Besier TF. 2005. Estimation of muscle forces and joint moments using a forward-inverse dynamics model. Medicine and Science in Sports and Exercise. 37(11):1911–1916.

Challis J. 1997. Producing physiologically realistic individual muscle force estimations by imposing constraints when using optimization techniques. Medical Engineering and Physics. 19(3):253–261.

Challis J, Kerwin D. 1993. An analytical examination of muscle force estimations using optimization techniques. Proceedings of the Institution of Mechanical Engineers Part H, Journal of engineering in medicine. 207(3):139–148.

Crowninshield R, Brand R. 1981. A physiologically based criterion of muscle force prediction in locomotion. Journal of Biomechanics. 14(11):793–801.

Damsgaard M, Rasmussen J, Christensen S, Surma E, de Zee M. 2006. Analysis of musculoskeletal systems in the AnyBody Modeling System. Simulation Modelling Practice and Theory. 14(8):1100–1111.

Delp S, Anderson F, Arnold A, Loan P, Habib A, John C, Guendelman E, Thelen D. 2007. OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE transactions on bio-medical engineering. 54(11):1940–50.

Dempster W. 1955. Space requirements of the seated operator: geometrical, kinematic, and mechanical aspects of the body, with special reference to the limbs.

Dumas R, Chèze L, Verriest J. 2007. Adjustments to McConville et al. and Young et al. body segment inertial parameters. Journal of biomechanics. 40(3):543–53.

Dumas R, Moissenet F, Lafon Y, Cheze L. 2014. Multi-objective optimisation for musculoskeletal modelling: application to a planar elbow model. Proceedings of the Institution of Mechanical Engineers Part H, Journal of engineering in medicine. 228(10):1108–13.

Dumont G. 1995. The Active Set Algorithm for Solving Frictionless Unilateral Contact Problems. In: Contact mechanics. p. 263–266.

Featherstone R. 2008. Rigid Body Dynamics Algorithms.

Flash T, Hogan N. 1985. The coordination of arm movements: an experimentally confirmed mathematical model. The Journal of neuroscience : the official journal of the Society for Neuroscience. 5(7):1688–1703.

Forster E, Simon U, Augat P, Claes L. 2004. Extension of a state-of-the-art optimization criterion to predict co-contraction. Journal of biomechanics. 37(4):577–581.

Happee R. 1994. Inverse dynamic optimization including muscular dynamics, a new simulation method applied to goal directed movements. Journal of Biomechanics. 27(7):953–960.

Herzog W. 1987. Individual muscle force estimations using a non-linear optimal design. Journal of Neuroscience Methods. 21(2-4):167–179.

Hill A. 1938. The Heat of Shortening and the Dynamic Constants of Muscle. Proceedings of the Royal Society B: Biological Sciences. 126(843):136–195.

Klein Horsman M, Koopman H, van der Helm F, Prosé L, Veeger H. 2007. Morphological muscle and joint parameters for musculoskeletal modelling of the lower extremity. Clinical biomechanics. 22(2):239–47.

Manal K, Gonzalez RV, Lloyd DG, Buchanan TS. 2002. A real-time emg-driven virtual arm. Computers in biology and medicine. 32(1):25–36.

Maurel W, Thalmann D, Hoffmeyer P, Beylot P, Gingins P, Kalra P, Thalmann NM. 1996. A biomechanical musculoskeletal model of human upper limb for dynamic simulation. Computer Animation and Simulation'96. (1):121–136.

Murai A, Kurosaki K, Yamane K, Nakamura Y. 2010. Musculoskeletal-see-through mirror: Computational modeling and algorithm for whole-body muscle activity visualization in real time. Progress in Biophysics and Molecular Biology. 103(2-3):310–317.

Pedotti A, Krishnan VV, Stark L. 1978. Optimization of muscle-force sequencing in human locomotion. Mathematical Biosciences. 38(1-2):57–76.

Pennestri E, Stefanelli R, Valentini PP, Vita L. 2007. Virtual musculo-skeletal model for the biomechanical analysis of the upper limb. Journal of Biomechanics. 40(6):1350–1361.

Pontonnier C, Dumont G. 2009. Inverse Dynamics Method using Optimisation Techniques for the Estimation of Muscle Forces Involved in the Elbow Motion. International Journal on Interactive Design and Manufacturing (IJIDeM). 3:227–235.

Pontonnier C, Dumont G. 2010. Interpolating Muscle Forces in an Inverse Dynamics Approach. In: Symposium on Computer Methods in Biomechanics and Biomedical Engineering.

Rasmussen J, Damsgaard M, Voigt M. 2001. Muscle recruitment by the min/max criterion - A comparative numerical study. Journal of Biomechanics. 34(3):409–415.

Rengifo C, Aoustin Y, Plestan F, Chevallereau C. 2010. Distribution of forces between synergistics and antagonistics muscles using an optimization criterion depending on muscle contraction behavior. Journal of biomechanical engineering. 132(4):041009.

Sartori M, Reggiani M, Farina D, Lloyd DG. 2012. Emg-driven forward-dynamic estimation of muscle force and joint moment about multiple degrees of freedom in the human lower extremity. PloS one. 7(12):e52618.

Siemienski A. 1992. Soft saturationan idea for load sharing between muscles. Application to the study of human locomotion. Biolocomotion: a century of research using moving pictures Promograph, Rome:293–303.

Song D, Lan N, Loeb G, Gordon J. 2008. Model-based sensorimotor integration for multi-joint control: Development of a virtual arm model. Annals of Biomedical Engineering. 36(6):1033–1048.

van den Bogert A, Geijtenbeek T, Even-Zohar O, Steenbrink F, Hardin E. 2013. A real-time system for biomechanical analysis of human movement and muscle function. Medical & biological engineering & computing. 51(10):1069–77.

Van Der Helm F. 1994. A finite element musculoskeletal model of the shoulder mechanism. Journal of Biomechanics. 27(5):551–569.

Zajac F. 1989. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. Critical reviews in biomedical engineering. 17(4):359–411.